Guided Tour Introduction

Welcome to CodeWarrior.

This Guided Tour is designed to introduce new users to the CodeWarrior Integrated Development Environment (IDE).

Using the Controls

Use these three controls to move through this tutorial.

Previous

Click to move to the previous page.

Return to Topics

Click to return to the main topics page.

Next

Click to move to the next page.

Adobe Acrobat

The standard Adobe Acrobat commands are also available on the View menu.

Guided Tour Overview

The topics covered in this overview of the Guided Tour include:

- Using the Controls
- Guided Tour Topics
- Required Knowledge
- Intended Audience
- Taking the Tour
- Host Conventions
- Where Else to Go Tutorials
- Where Else to Go Documentation

Guided Tour Topics

This tutorial introduces the CodeWarrior IDE features. This tutorial includes:

- STAGE 1: Launching the CodeWarrior IDE
- STAGE 2: Setting IDE Preferences
- STAGE 3: Opening a CodeWarrior Project
- STAGE 4: Touring the Project Window
- STAGE 5: Using the CodeWarrior Editor
- STAGE 6: Finding Text
- STAGE 7: Setting Target Options
- STAGE 8: Building Projects with CodeWarrior
- STAGE 9: Touring the Browser
- STAGE 10: Touring the Debugger

Required Knowledge

This tour assumes that you have a basic understanding of using your computer system. If you are just getting started with computers, refer to the user guide that came with your computer system. Once you feel comfortable with using your computer, come back and start again.

NOTE: The CodeWarrior tutorials are designed to teach you how to use CodeWarrior to successfully create, compile, and debug executable code. These tutorials will not teach you how to program a computer. If you wish to learn programming, visit your local bookstore for beginning programming books, see your local community college for classes on programming, or refer to the 3rd-party documentation that may be provided on your CodeWarrior CD.

Intended Audience

The Guided Tour (Basic) tutorial is intended for all first-time code warriors. If you have not already taken the Guided Tour tutorial, we recommend that you take a few minutes to go through that tutorial before continuing here. The Guided Tour will give you an overview of the entire CodeWarrior IDE.

Beginning Programmers

If you are just beginning your career as a programmer then you are in for a treat. The CodeWarrior environment provides all the tools you need for any programming task, from simple applications to advanced professional-quality programs. Go through this tour, then follow the recommendations at "Where Else to Go Tutorials" to continue your explorations of CodeWarrior.

more…

Intended Audience (continued)

Experienced Programmers

After taking the Guided Tour tutorial, experienced programmers new to CodeWarrior should refer to this and the other tutorials for in-depth knowledge about the various components of the CodeWarrior environment. Many CodeWarrior Targeting manusl include basic tutorials that demonstrate how to use the tools to program for a particular platform.

Taking the Tour

The CodeWarrior tutorials are designed to help you become familiar with the CodeWarrior IDE. This tutorial consists of a series of stages. Each stage, in turn, is a series of slides, like this one. Each slide describes a feature or a task you can perform. You will gain the most benefit from these tutorials if you follow along with the CodeWarrior IDE while reading and performing the actions described in the tutorial.

To Use a Tutorial:

1. In Acrobat Reader 3.0, read the text on a slide.
2. If actions are described (example: Choose Open from the File menu), then switch to the CodeWarrior IDE.
3. In the CodeWarrior IDE, perform the action.
4. Switch back to Adobe Acrobat Reader, read the next action.

NOTE: Both Adobe Acrobat Reader and the CodeWarrior IDE must be running.

Host Conventions

Hosts: While the CodeWarrior IDE runs on several different host platforms, the functionality on these host platforms is not always identical. There are occasions when a feature is implemented for one or more hosts and not for others. This tutorial only demonstrates features found on all supported hosts.

Pictures: The IDE's visual interface on these hosts is nearly identical in all significant respects. When discussing a particular interface element such as a dialog box or window, the screenshot may come from any host. If you are using CodeWarrior on a different host, you should have no difficulty understanding the picture.

Keyboard: The default keyboard shortcuts for CodeWarrior on these platforms are very similar. However, keyboards and shortcuts do vary across host platforms. For example, a typical keyboard for a Windows machine has an Alt key, but a typical keyboard for a Mac OS computer calls the same key the Option key

more…

Host Conventions (continued)

To handle these kinds of situations, CodeWarrior documentation identifies and uses the following paired terms in the text:

- `Enter/Return` - the “carriage return” or “end of line” key. This is not the numeric keypad Enter key, although in almost all cases that works the same.
- `Backspace/Delete` - the key that (in text editing) causes the character before the insertion point to be erased (not to be confused with `Delete/Del`, the “forward delete” key)
- `Ctrl/Command` - in most cases, CodeWarrior maps the Windows Ctrl key the same way as the Mac OS Command key.
- `Alt/Option` - in most cases, CodeWarrior maps the Windows Alt key the same way as the Mac OS Option key.

For example, you may encounter instructions such as “Press `Ctrl/Command+S` to save a file,” or “Alt/Option click the Function pop-up to see the functions in alphabetical order." Use the appropriate key as it is labeled on your keyboard.

Where Else to Go Tutorials

Once you have finished this tutorial, be sure to explore the other CodeWarrior tutorials, and the documentation that accompanies this product. For an overview of the IDE, see:

- Guided Tour–overview of the CodeWarrior IDE
- Project Manager (Basic)–introduces the Project Manager
- Editor (Basic)–introduces the CodeWarrior Editor
- MW Debugger (Basic)–introduces debugging features
- Find & Replace–introduces Find and Replace features

Where Else to Go Documentation

To learn more about programming for a particular target operating system or processor, see the corresponding CodeWarrior Targeting manual. The CodeWarrior Guided Tour features are also documented in:

- IDE User Guide–documents IDE components, including Find and Replace.
- Debugger User Guide–documents the Metrowerks standalone debugger.
- Targeting manual for your programming target. Examples include Targeting Windows, Targeting Rhapsody, and Targeting Palm OS.

Getting Started

With all of the preliminaries are out of the way, it's time to explore the CodeWarrior IDE.

Launching the CodeWarrior IDE

In this stage you learn how to easy it is to launch the CodeWarrior IDE, opening the gateway to a powerful set of programming tools, and beginning your career as a code warrior.

- Launch the CodeWarrior IDE on Windows
- Launch the CodeWarrior IDE on Mac OS

Launch the CodeWarrior IDE on Windows

Before you can begin learning about the CodeWarrior IDE, it's necessary to launch the IDE.

To Launch the CodeWarrior IDE on Windows:

1. Click the Start button, click Programs, click Metrowerks CodeWarrior, then click the CodeWarrior IDE item.

   The CodeWarrior application launches, and the CodeWarrior menus appear at the top of the screen.

Launch the CodeWarrior IDE on Mac OS

If you plan on running the CodeWarrior IDE on a Mac OS platform, follwo these steps.

To Launch the CodeWarrior IDE on Mac OS:

1. Locate the CodeWarrior IDE.
2. Double-click the application icon.

   The CodeWarrior application launches, and its menus appear.

Summary

This concludes the launching CodeWarrior stage of this tour.

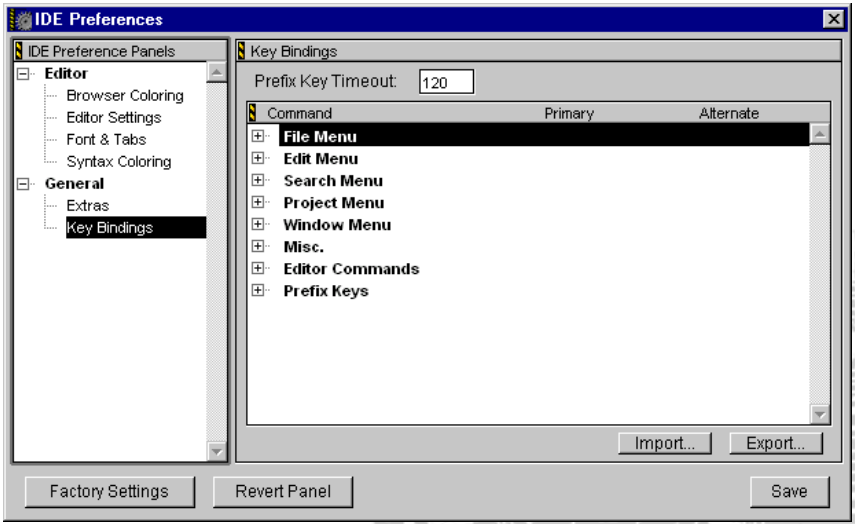
In this stage you learned how to:

- Launch the CodeWarrior IDE

In the next stage you'll learn about the preference setting panels within CodeWarrior.

Setting IDE Preferences

With the CodeWarrior IDE up and running, let's introduce you to the Preferences window.



Preferences Overview

The Preferences window enables you to set options that are global to all projects you will work with in the CodeWarrior IDE.

The topics in this stage include:

- Opening the Preferences Window
- Touring the Preferences Window
- Viewing a Preference Panel
- Closing the Preferences Window

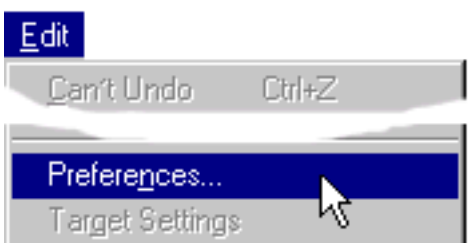
Opening the Preferences Window

To change a CodeWarrior preference, the Preferences window must be open.

To Open the Preferences window:

- Choose **Preferences** from the **Edit** menu.

  The Preferences window appears on the screen.



Touring the Preferences Window

Let's take a moment to identify some of the components found in the Preferences window.

Preference Window Views

IDE Preference Panels–lists all of the preference panels that globally affect the CodeWarrior environment. Changes made in these panels are reflected in all new projects you create afterward.

Preference Panel view– displays the panel currently selected in the Preference Panel list (shown as Editor Settings).

NOTE: The individual preference panels are fully discussed in other Metrowerks tutorials as they become relevant to the discussion, and in the IDE User Guide.

Preferences Window Controls

The Preferences window also includes three controls that assist you in setting the various preference panel options. These controls include:

Factory Settings–resets the current panel to the default settings. Using Factory Settings only affects the current panel's settings. This control is useful should you wish to restore the panel settings to the original CodeWarrior values.

Revert Panel–restore the current panels settings to the state they were in when the panel was first opened. This control is useful when you make changes to a panel, then decide the original settings were just fine.

Save–commit any changes made in the preference panel. Once you're satisfied with the settings in a panel, click Save to commit to the changes.

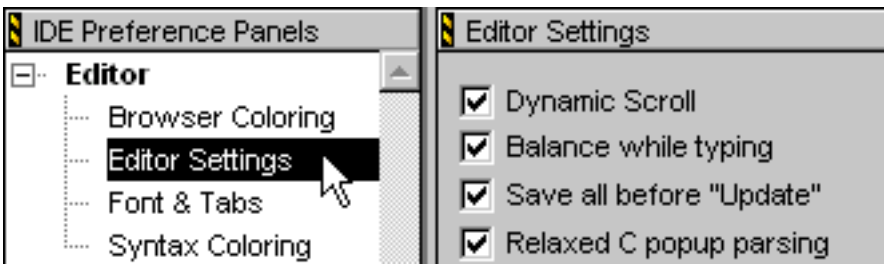
Viewing a Preference Panel

There are several different preference panels available in the Preference Panels list. To set a particular preference, you must choose a panel.

To View a Preference panel:

- Click the panel name in the Preference Panel list, **OR**
  Use the Up/Down arrow keys to change the selection.

  The selected panel name is highlighted in the Preference Panel List and the panel appears to the right of the list.



Closing the Preferences Window

Once you finish setting your global preferences, just close the window.

To Close the Preferences window:

- Click the Preference window's close box, **OR**
  Choose **Close** from the **File** menu, **OR**
  Type the keyboard shortcut Ctrl/Command+W.

  The Preferences window closes.

Summary

This concludes the Preference window stage of the tour.

In this stage you learned how to:

- Open the Preferences window
- Identify Preference window components
- Display a specific preference panel
- Close the Preferences window

Now that you understand the use of the Preferences window, let's explore how to open a CodeWarrior project.

Opening a CodeWarrior Project

In this stage you learn how to open a CodeWarrior project.

A project is a collection of files that the CodeWarrior compiler and linker use to create computer code. Examples of compiled code include applications, libraries, precompiled headers, shared libraries on the Mac OS, and dynamic linked libraries on Windows.

The topics covered in this stage include:

- Opening a CodeWarrior Project

Opening a CodeWarrior Project

Open a project file in exactly the same way you open any other file from within an application on your computer. In this step you open the `gnu_go.mcp` project file.

To Open a Project:

1. Choose **Open** from the **File** menu, **OR**
   Type the keyboard shortcut Ctrl/Command+O.
2. Locate and select the Project File.

   For this tour, choose `gnu_go.mcp`. You can find this file in the `Source Examples` folder in the `Core Tutorials` folder on the CD.
3. Click **Open**.

   The project file opens and a Project window appears.

Summary

This concludes the Open a Project portion of this tour.

In this stage you learned how to:

- Open a CodeWarrior project file

Now that you have a project open, let's explore the features of the Project window.

For More Information

See the Core Tutorial: Project Manager (Basic)

Touring the Project Window

To effectively use the CodeWarrior IDE, you should be familiar with the Project window. The Project window displays lots of information about a project, so identifying and understanding that information is our next step. The Project window includes:

- View Tabs
- Toolbar
- Toolbar Tooltips
- Project Window Columns
- Groups and Disclosure Triangles
- Interface Pop-ups
- Context Pop-ups

View Tabs

The Project window uses view tabs to present information in a uniform manner. Click a view tab to change the information displayed in the Project window.

The Project window uses these view tabs:

Files–enables you to inspect file information in a project.

Link Order–enables you to specify the order in which files are compiled.

Targets– enables you configure the targets a project will build and the order in which the targets are built.

NOTE: The Link Order view is known as the Segment view when working with Macintosh 68K projects.

Toolbar

The Toolbar provides icon buttons that access most of the CodeWarrior IDE features. The toolbar appears in two formats:

Window Toolbar–present in several windows within CodeWarrior, including the Project window.

Floating Toolbar–appears as a floating palette.

Hiding and Showing Toolbars

You specify whether a toolbar is visible or not using the menu commands located on the **Toolbar** submenu of the **Window** menu.

To Show a Toolbar:

- Choose **Show Window Toolbar** to make the Window Toolbar visible.
- Choose **Show Floating Toolbar** to make the Floating Toolbar visible.

To Hide a Toolbar:

- Choose **Hide Window Toolbar** to make the Window Toolbar invisible.
- Choose **Hide Floating Toolbar** to make the Floating Toolbar invisible.

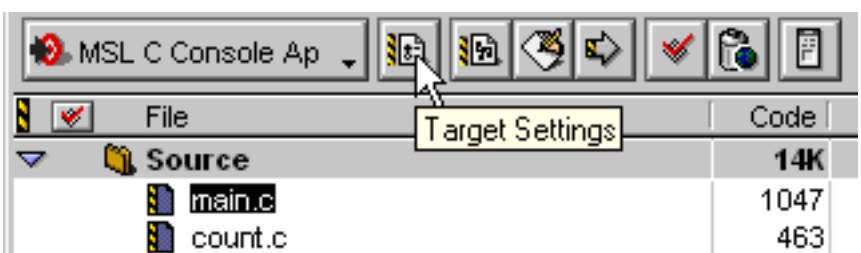
Toolbar Tooltips

Use Tooltips to identify icons on a toolbar.

To View a Tooltip:

1. Press the Control key.
2. Position the cursor over a toolbar icon.

   The tooltip appears describing the icon underneath the cursor.



Project Window Columns

The columns that appear in the Project window's File view include:

Touch–indicates whether a file needs to be compiled.

File–identifies the files and groups in a project.

Code– shows the size of the compiled executable object code for each file.

Data–shows the size of the non-executable data area for each file.

Debug–indicates whether debugging information is being generated by the file.

Targets–indicates whether a file belongs to the current target.

Checkout Status–indicates the current checkout status of a file from a version control system.

Link–located only in the Targets view, indicates the link status of a particular subtarget to a target.

Groups and Disclosure Triangles

Groups

Groups provide a way to organize project files into logical categories to help you keep track of your files. Groups have no effect on the binary code produced by the compiler, only on their organization within the Project window. Groups can be rearranged, added, or deleted from a project.

Disclosure Triangles

Click a disclosure triangle to the left of the group name to expand or collapse a group.

Interface Pop-ups

Interface Pop-ups reveal additional information about the group or file associated with it. The two types of interface pop-ups include:

Group's Interface Pop-up–displays a list of all files contained in that group.

File's Interface Pop-up–displays a list of interface files required to compile the file, and allows you to 'touch' the file, forcing it to recompile on the next build.

To View an Interface Pop-up:

- Click on the pop-up menu to the right of the file or group name.

  The file or group interface pop-up appears.

Context Pop-ups

The two forms of Context Pop-ups available in the Project window include:

Group Context Pop-up–enable you to Check Syntax, Preprocess, Precompile, Compile, Disassemble, or Remove Selected Items.

File Context Pop-up–does everything the Group Context Pop-up does as well as showing the File Path, and Reveal in Finder the file's location.

To View a Context Pop-up:

1. Position the cursor over a file or group name.
2. Click and hold the mouse button down.

   The File context pop-up appears.

Summary

This concludes the Project Manager stage of our tour.

In this stage you learned how to:

- Use the File, Link, and Target view tabs
- Identify the Window and Floating Toolbars
- Use tooltips
- Identify the columns that appear in the File, Link, and Target views
- Identify groups and disclosure triangles
- Access the different pop-up menus used in the Project window

Now that you have a better understanding of the Project Manager, let's move onto the next stage and tour the CodeWarrior Editor.

Using the CodeWarrior Editor

The CodeWarrior Editor window is where you will spend most of the time creating source code for your project. This stage of the Guided Tour introduces you to simple operations in the Editor.

The topics covered in this stage include:

- Opening a File
- Entering Text into a File
- Saving Changes to a File
- Examine Editor Preferences

Opening a File

There are a variety of methods you can use to open a file from within a project. These methods include:

- Opening a File from the Project Window
- Opening a File with a Group Pop-up
- Opening a File from the File Menu

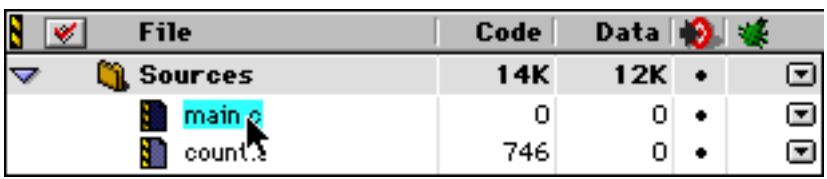
Opening a File from the Project Window

You can open any editable text file listed in the Project window. In this step you open the `main.c` file from the `gnu_go.mcp` project.

To Open a File From the Project Window:

- Double-click the file name in the Project window.

  The file opens and appears in its own window.



NOTE: You can also select a file in the Project window using the arrow keys and press the Enter/Return key to open it.

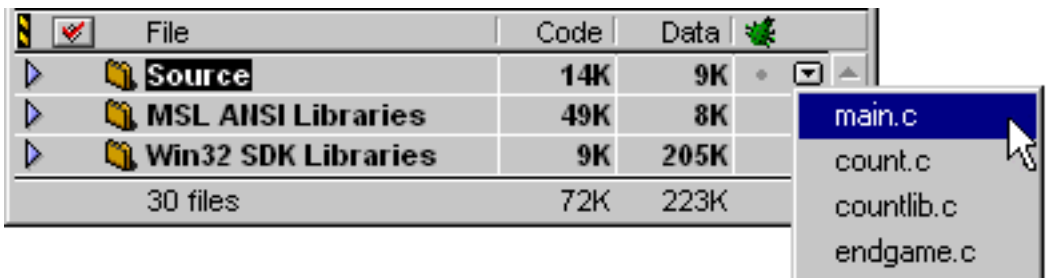
Opening a File with a Group Pop-up

You can also open an editable text file within a group by using the Group Pop-up that appears to the right of the group name.

To Open a File from a Pop-up Menu:

- Select the file from a group pop-up in the Project window.

  The file opens and appears in its own window.



Opening a File from the File Menu

You can use this method to open any text file, whether it is included in the project or not. In this step you open the file `endgame.c` using the **File** menu.

To Open a File from a Menu:

1. Choose **Open** from the **File** menu, **OR**
   Type the keyboard shortcut Ctrl/Command+O.
2. Locate the file `endgame.c` to open.
3. Select the file.
4. Click **Open**.

   The `endgame.c` file opens and appears in its own window.

Entering Text into a File

Now that you have a file open in the editor, you can enter text. The CodeWarrior Editor operates like a regular text editor, with many special features designed to ease navigation. The Editor (Basic) tutorial introduces you to the Editor's special features. In this step you make changes made to the `endgame.c` file.

To Enter Text:

1. Position the cursor at the point where you want to insert text.
2. Click the mouse button.

   An I-beam cursor appears at the text insertion point.
3. Begin typing. Try "`// This is a test!`".

   Your keystrokes appear at the text insertion point.

Saving Changes to a File

As you modify your source files, its a good idea to save the changes to disk to prevent data loss. In this step you save changes made to the `endgame.c` file.

To Save a File:

- Choose **Save** from the **File** menu, **OR**
  Type the keyboard shortcut Ctrl/Command+S.

  CodeWarrior updates the file and Save is disabled.

TIP: Monitor the Dirty File Marker in the editor windows to know when a file needs saving.

Examine Editor Preferences

You can customize the Editor using the Preferences window.

To Customize the Editor:

1. Choose **Preferences** from the **Edit** menu.
2. Select **Editor Settings** from the preference panel list.
3. Change the desired Editor options.
4. Click **Save**.

   Your changes take effect immediately upon return to the Editor.

Summary

This concludes the CodeWarrior Editor stage of our tour.

In this stage you learned how to:

- Open a file
- Enter text
- Save modified files
- Open the editor settings

Now that you are comfortable with the CodeWarrior Editor, it's time to explore the Find & Replace features.

For More Information

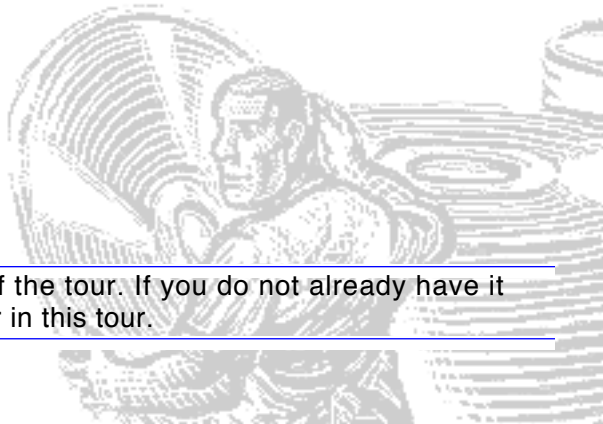
See the Core Tutorial: Editor (Basic)

Finding Text

As you work on source code, you will often discover a need to replace variables, functions, or class names with better terms. This portion of the Guided Tour introduces the search capabilities of the CodeWarrior IDE.

The topics include:

- Opening the Find Window
- Find Window Components
- Performing a Find
- To Repeat a Find
- Closing the Find Window



NOTE: Use the file `main.c` for this stage of the tour. If you do not already have it open, use the techniques you learned earlier in this tour.

Opening the Find Window

In order to start a search operation, you must enter a search string into the Find window's Find field. One way of doing this, is with the Find window. In this step you open the Find window.

To Open the Find window:

- Choose **Find** from the **Search** menu, **OR**
  Type the keyboard shortcut Ctrl/Command+F.

  The Find window appears on screen.

Find Window Components

To search files in CodeWarrior, the two components of interest in the Find window include:

Find field–enter text to search for in this field

Find button–click here to search the current source file for a match to the Find field's contents

Performing a Find

With the Find window open, you first enter the search string into the Find field. In this step you perform a Find operation.

To Enter a Search String:

1. Click in the **Find** field.
2. Enter a search string in the **Find** field.

   For this example, enter: `mymove`.
3. Click **Find**, **OR**
   Press the Enter/Return key, **OR**
   Type the keyboard shortcut F3/Command+G.

   CodeWarrior searches the current editing window starting at the current cursor position for a match to the search string. If a match is found, the text scrolls to the location and the Editor highlights the matching string.

To Repeat a Find

After you enter a search string, you can easily locate additional matches in the current editing window by repeating the search. In this step you repeat the search using the same settings.

To Repeat a Search:

- Click the **Find** button again, **OR**
  Enter the keyboard shortcut F3/Command+G.

  If another match is found, the text scrolls to the location and the Editor highlights the matching string.

Closing the Find Window

Once you have finished with the Find window, you can close the window to free up screen real estate for other purposes. In this step you close the Find window.

To Close the Find window:

- Click the window's close box, **OR**
  Choose **Close** from the **File** menu, **OR**
  Enter the keyboard shortcut Ctrl/Command+W.

  The Find window closes.

Summary

This completes the Find & Replace stage of this tour.

In this stage you learned how to:

- Open and close the Find window
- Perform a search operation
- Repeat a search
- Close the Find window

Now that you are familiar with CodeWarrior's Find & Replace features, it's time to explore Target Settings.

Setting Target Options

Target setting apply only to the current project. Settings in one project do not affect settings in another project. The Target Settings window appears very similar to the Preferences window, with many common features.

Setting Target Overview

Projects can contain one or more targets. For example, you can have a debug target and a production target. Every target in the project has its own unique settings. The Target Settings dialog box enables you to set target options on all of target setting panels.

This stage covers:

- Opening the Target Settings Window
- Target Settings Window Components
- Target Settings Window Controls
- Viewing a Settings Panel
- Closing the Target Settings Window

Opening the Target Settings Window

To change a target option, the Target Settings window must be open. In this step you open the Target Settings window.

To Open the Target Settings Window:

- Choose **<Target> Settings** from the **Edit** menu.

  The Target Settings window opens on the screen.

<Target> will be replaced by the name of the current target in the project file. On a PC running Windows for example, the targets include: `gnu_go x86`, and `go_debug`. On the MacOS the targets include: `gnu_go 68K`, `gnu_go PPC`, and `gnu_go FAT`.

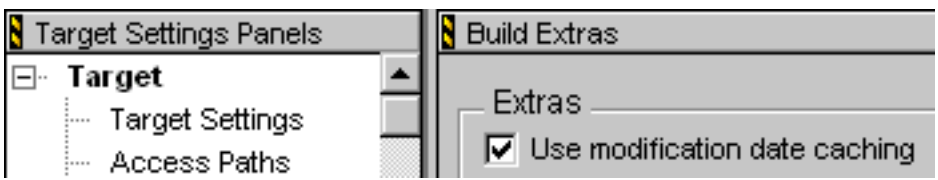
TIP: You can also double-click a target name in the Targets view to open the Target Settings window.

Target Settings Window Components

Let's take a moment to identify some of the components found in the Target Settings window. They include:

Settings Panels list–lists all of the settings panels that affect a CodeWarrior project. Changes made in these panels only affect the current project.

Settings Panel view–displays the currently selected settings panel in the Settings Panel list. Each panel has a different collection of settings.



NOTE: The individual settings panels are fully discussed in other Metrowerks tutorials as they become relevant to the discussion.

Target Settings Window Controls

The Target Settings window includes three controls that match those found in the Preferences window. These controls are located at the bottom of the Target Settings window and include:

- Factory Settings
- Revert Panel
- Save

NOTE: See Preferences Window Controls for more information on these controls.

Viewing a Settings Panel

There are several different preference panels available in the Target Settings list. To set a particular target option, you must choose a panel. In this step you view a specific settings panel.

To View a Settings panel:

- Click the panel name in the Target Settings list, **OR**
  Use the Up/Down arrow keys to change the selection.

  The selected preference panel name highlights and the panel appears to the right of the list.

Closing the Target Settings Window

Once you finish setting the target options, just close the window. In this step you close the Target Settings window.

To Close the Target Settings Window:

- Click the Target Settings window's close box, **OR**
  Choose **Close** from the **File** menu, **OR**
  Type the keyboard shortcut Ctrl/Command+W.

  The Target Settings window closes.

Summary

This concludes the quick tour of the Target Settings.

In this stage you learned how to:

- Open and close the Target Settings window
- Identify Target Settings window components
- Close the Target Settings window

Now that you can set Target Settings for projects, it's time to build a compiled application from the `gnu_go.mcp` project.

Building Projects with CodeWarrior

In this stage we'll introduce you to four methods of building a project, and show you how to set the target settings for a project to create a stand-alone application.

This tour of Building Projects covers:

- Project Building Methods
- Setting Target Options
- Building a Project

Project Building Methods

There are several methods you can use to compile a project. Depending upon your point in the development process, you will definitely use one, or more of these methods to create your projects.

The four building methods include:

- Updating a Project
- Making a Project
- Running a Project
- Debugging a Project

Updating a Project

Use **Bring Up To Date** on the **Project** menu to compile all modified or "touched" files. This process creates object code for compiled files, but does not link the code into a binary file.

Making a Project

Use **Make** on the **Project** menu to link a project into a binary file. Make only creates binary files on the hard disk, it does not interactively run the final application.

Running a Project

Use **Run** on the **Project** menu to launch the application created by your project. If the project type is not an application, the **Run** command is disabled. **Run** will update and link the code if necessary.

Debugging a Project

Use **Debug** on the **Project** menu to launch the application under control of the Debugger. You can then debug your code. **Debug** will also update and link the code if necessary.

Setting Target Options

In this step you name the output application for the `gnu_go.mcp` project.

To Set an Application Name:

1. Choose **x86 Target** in the settings panel list for Windows, **OR**
   Choose **68K Target** for the Mac OS.
2. Choose **Application** in the **Project Type** pop-up menu.
3. Enter an application name in the **File Name** field.

   For example, use `gnu_go.exe` for a Windows application, or just `gnu_go` for a Mac OS application.
4. Click **Save.**

   When the alert appears asking to re-build the project, click **OK**.

NOTE: The name of <Target> will change depending upon the target chosen in the Project window.

Building a Project

With the project information in place, it is now time to actually build the project. In this step you build the `gnu_go.mcp` project.

To Build an Application:

- Choose **Run** from the **Project** menu, **OR**
  Type the keyboard shortcut Shift+F5/Command-R.

  The project compiles, then the `gnu_go` application launches.

Summary

This concludes the Build Project stage of the tour.

In this stage you learned how to:

- Identify the different methods of compiling
- Setup a project
- Build a project

Phew! You've covered a lot of ground to reach this point. You now know the Project and Editor windows, how to set Preferences and Target Settings, and can build source files into a stand-alone application. Before we finish, let's explore two more features of CodeWarrior, namely the Browser and the Debugger that are used to navigate and debug your code respectively.

Touring the Browser

This stage introduces you to the Metrowerks Browser, a tool for examining project source code from different perspectives. In addition, the Browser works with both procedural and object-oriented code, and with the C, C++, Object Pascal, and Java compilers.

Browser Overview

The Browser stores a database of all the symbols in a project. A symbol is any special item created in the code, such as function names, variables, constants, and so forth. The Browser provides several user interface views that allow quick and easy access to the project's symbol data, regardless of language.

The Browser topics include:

- Activating the Browser
- Browser Views
- Browser Context Pop-up Menu

Activating the Browser

Before you can use the Browser database, CodeWarrior must be set to create the browser information. In this step you activate the Browser for the `gnu_go.mcp` project.

To Activate the Browser:

1. Choose **Target Settings** from the **Edit** menu, **OR**
   Click the **Target Settings** icon in a Toolbar.
2. Select the **Build Extras** settings panel.
3. Enable the **Activate Browser** checkbox.
4. Click **Save.**
5. Choose **Bring Up To Date** from the **Project** menu.

   The Browser activates and the compiler generates the database.

Browser Views

The Browser provides different methods of viewing project information. You access the different Browser views using the **Show Catalog Window**, **Show Hierarchy Window**, and **New Class Browser** menu items found on the Window menu. The three Browser views include:

Catalog window–contains a comprehensive view of all browser symbols including classes, constants, enumerations, routines, global variables, routine templates, and type definitions.

Hierarchy windows–provide graphical views of the class hierarchy.

Class Browser windows–provide class-based views of browser symbols.

NOTE: The project `gnu_go.mcp` does not contain any classes to view, so the Hierarchy and Class Browser windows will remain blank.

Browser Context Pop-up Menu

Use the Context Pop-up menu to quickly link a symbol to its declaration, or to all routines where it is located or used within the project. In this step you open the Catalog window, and choose Functions to use the following steps.

To Access a Context Pop-up:

1. Position the cursor over a symbol in the code (example: `bfslist`).
2. Click and hold the mouse button down.

   A pop-up menu appears listing relevant commands.
3. Select an item (example: **Go to Definition** or **Find all Implementations**).

   An editor window opens and displays the chosen item.

TIP: To bypass the pop-up delay settings, press the Control key.

Summary

This concludes the Browser portion of this tour.

In this stage you learned how to:

- Activate the Browser
- Identify the three Browser windows
- Use the Browser pop-ups

We're almost done. The final stage to explore is the Metrowerks Debugger. A separate application that enables you to interactively run and debug your compiled code.

Touring the Debugger

This stage introduces you to the Metrowerks Debugger, a source-level debugger for the CodeWarrior IDE.

The Debugger is the final critical component of the CodeWarrior IDE, and this is the final stage of the Guided Tour.

Debugger Overview

It's not always possible to write source code that operates correctly the first time it is compiled and linked. That is where the Debugger is essential. It allows you to watch source code execute line-by-line, view and edit variables, and locate problems in the program.

This tour of the Debugger covers:

- Enabling the Debugger
- Running the Debugger
- Common Debugger Windows
- Disabling the Debugger

Enabling the Debugger

Before you can use the Metrowerks Debugger, the CodeWarrior IDE must generate debugging information for the project. You tell CodeWarrior to do this by enabling the Debugger. In this step you enable the Debugger for the `gnu_go.mcp` project.

To Enable the Debugger:

- Choose **Enable Debugger** from the **Project** menu.

  Markers are placed in the Debug column of the Project window for all source files in the project.

  On Windows, the **Enable Debugger** menu item on the **Project** menu changes to **Disable Debugger.**

  On the Mac OS, the **Enable Debugger** and **Run** menu items on the **Project** menu change to **Disable Debugger**, and **Debug** respectively. In addition, .

Running the Debugger

With the CodeWarrior IDE ready to generate debugger information for the project, its time to run (or debug) the project. In this step you **Run** the Debugger from the CodeWarrior IDE.

To Run the Debugger:

- Choose **Debug** from the **Project** menu, **OR**
  Type the keyboard shortcut Shift+F5/Command+R.

  The project is compiled and linked, then the Debugger is launched (if not already running) with the application active and ready to debug.

Common Debugger Windows

The Metrowerks Debugger uses many different windows to facilitate debugging. The three most commonly used windows include:

Browser window–displays debugging information for all the files in the project and allows you to view all the files and global variables.

Program window–displays debugging information about the source code file containing the currently running routine.

Debugger toolbar–provides buttons to execute commands on the Control menu. The Debugger Toolbar also appears within the Program window.

Disabling the Debugger

Once you are finished using the Debugger, you should disable it to compile your final project without debugging information. In this step you disable debugging the `gnu_go.mcp` project.

To Disable the Metrowerks Debugger:

- Choose **Disable Debugger** from the **Project** menu.

  On Windows, markers are removed in the Debug column of all source files in the project. In addition, the **Disable Debugger** menu item on the **Project** menu changes to **Enable Debugger**.

  On the Mac OS, markers are removed in the Debug column of all source files in the project. In addition, the **Disable Debugger** and **Debug** menu items on the **Project** menu change to **Enable Debugger**, and **Run** respectively.

Summary

This concludes the Metrowerks Debugger stage of the tour.

In this stage you learned to:

- Enable the Debugger
- Run the Debugger
- Identify Debugger components
- Disable the Debugger

For More Information

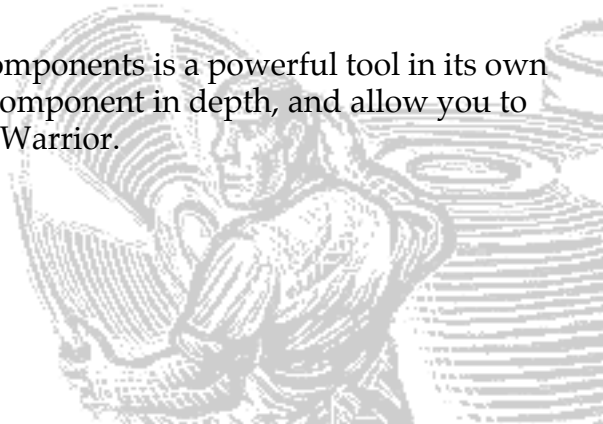
See the Core Tutorial: MW Debugger (Basic)

Conclusion

Congratulations!

You have completed the CodeWarrior Guided Tour. You have met all the major components of the CodeWarrior IDE: the Project Manager, the Editor, the Debugger, as well as the Preferences, Target Settings, and Find windows. While there is still a lot left to learn, you are well on your way to becoming a successful code warrior.

This is only the beginning. Each of the components is a powerful tool in its own right. Additional tutorials explain each component in depth, and allow you to explore the power and potential of CodeWarrior.



Technical Support

If you should encounter any difficulties with this guided tour, contact Metrowerks Technical Support for assistance. They can be reached at:

`support@metrowerks.com`

Copyright Information

Metrowerks CodeWarrior copyright ©1993–1998 by Metrowerks Inc. and its licensors. All rights reserved.

Documentation stored on the compact disk(s) may be printed by licensee for personal use. Except for the foregoing, no part of this documentation may be reproduced or transmitted in any form by any means, electronic or mechanical, including photocopying, recording, or any information storage and retrieval system, without permission in writing from Metrowerks Inc.

Metrowerks, the Metrowerks logo, CodeWarrior, and Software at Work are registered trademarks of Metrowerks Inc. PowerPlant and PowerPlant Constructor are trademarks of Metrowerks Inc.

All other trademarks and registered trademarks are the property of their respective owners.

ALL SOFTWARE AND DOCUMENTATION ON THE COMPACT DISK(S) ARE SUBJECT TO THE LICENSE AGREEMENT ON THE CD.

Revision Code: 1/14/98-lft